- TristanMinPerturbNotes
- TristanAthleteNotes (moved to a4o site just in case I put anything sensitive there)
- ResourceProfileImprovements
- ResourceNotes
- DynamicEuropaNotes

Misc Thoughts/Questions/Notes

- Is Trac currently backed up?
- Combine .cfg files that results from makeproject into one file?
- Should all of the potential includes be in our skeleton .nddl file(eg Resources.nddl)?
- Nddl desired based on ATHLETE modeling:
    - ♦ Ability to refer to member tokens of another token (ie ability to define member tokens). This would help avoid creating of lots of slaves that I know need to be merged with other tokens.
    - ♦ Containers (vector, for example), to ease the 6 leg pieces etc...
- Think HARD about what visualization features would help the debugging I'm doing, so I can add them later!

Doucmentation Updates

Big pieces missing:

- Various items in the EuropaDocs list still don't even have links
- EuropaExtensions
- NddlCfg
- BuildingOwnSolver
- ResourceSearchNotes

Smaller pieces missing:

- ArchitectureOverview is somewhat out of date (missing PSUI stuff, for example)
- DocGlossary needs work, especially links to other places (probably needs more terms added too?)
- PlanningApproach has a section (Dyamic Objects) that has a stub, but no information.

Also:

- This re-org might be more helpful:
    - ♦ Move architecture overview from User Documentation to the Overview section
    - ♦ Reorganize the User Documentation section as a 'how to get things done' page instead of a 'how europa works page'
        - ◊ Section for modeling (nddl, constraints, own constraint)
        - ◊ Section for solving (modifying solvers, own solver, etc)
        - ◊ Section for visualizing
        - ◊ Section for debugging
        - ◊ Section for embedding
- Here's a potential page to add to the EUROPA components section of the main EuropaDocs page:
    - ♦ Temporal Network
- Should we have even tighter integration between architecture overview and subsequent docs?

- There are a few links that don't have a good place to go (Michael's resource search notes, for example)
- Jeremy really likes the APEX documentation which I've looked through. Thoughts on that:
    - ♦ The high-level difference appears to be in the philosophy of the user docs organization. They have a big 'Using APEX' section where each subsection describes how to use a given piece, whereas our big section is more of a 'description of our pieces' rather than a 'how to use our pieces'.
    - ♦ There are certainly some pieces we might want to include like a Troubleshooting section.
    - ♦ The APEX docs are written as more of a standard user manual whereas we've written a wiki website, and that may account for other differences. I don't feel strongly that we need to mimic the APEX documentation.

FAQ fodder (ie helpful things I've learned)

- Running jam on my project gives "isValid() is false" or something like that. Reason is a variable somewhere referring to a class for which I haven't created any instances.
- TODO: Record common error messages and what they really mean:
    - ♦ Error: dom.intersects(baseDomain()) is false -> Means there is a constraint here that isn't allowed given current domain (for example, setting a variable to one object when it was already set to another!)
- Predicate variables aren't bound unless they are member variables (ie in declaration), at least by default.
- The following happens if you refer to a class member variable in one of it's tokens without preceding it with 'object.' Matthew may try to fix this:

```
toyProject-model.cc: In member function ?virtual void NDDL::YourObject$helloWorld$0$0::handleExe
toyProject-model.cc:342: error: expected type-specifier before ?null?
toyProject-model.cc:342: error: expected `)' before ?null?
toyProject-model.cc:342: error: no matching function for call to ?NDDL::YourObject$helloWorld$0$
```

- This happens if you try to have variables of the built-in resource types rather than of a class that inherits from the built-in types (see #137)

```
toyProject-model.cc: In member function ?virtual void NDDL::YourObject$helloWorld$0$0::handleExe
toyProject-model.cc:876: error: expected type-specifier before ?Resource?
toyProject-model.cc:876: error: expected `)' before ?Resource?
toyProject-model.cc:876: error: no matching function for call to ?NDDL::YourObject$helloWorld$0$
```

- Errors can occur if you don't have the right Resource-related stuff in NDDL.cfg. This is an issue for compiled code, but probably not for interpreter.
- unexpected token: UnaryResource ... if resource-related, need #include "Resources.nddl"
- Don't confuse equal/equals with eq (they're for different things)
- foreach doesn't work as I expected (ie foreach leg that's not this one doesn't work unless this one specified already!)
- Easy to create problem where no tokens need to be within horizon and solver won't do anything!
- With interpreter, can be easy to run without realizing that there's a problem in nddl code (ie it continues past it and
- If java code created, need to run `rm dist/europa/lib/*.jar; rm build/lib/*.jar` in order for build to recompile (ugh)
- makeproject uses files in dist, not the original files so be careful!
- Don't use 'set(x)' to set a variable to a value. Instead use intersect(x,x). The former is meant to solely be used internally!
- internal jam targets need PLASMA_HOME defined, while EUROPA_HOME only is needed for everything else. Using EUROPA_HOME for jam stuff doesn't work since it's unrelated to jam tree :).

- If you 'activate' in the initial state, be careful. For example, I got weird results when I activated and then set the start to be what I wanted LATER in the initial state file.
- In nddl, be sure to use floats even if integer values. For example, super(1.0, ...) didn't work for me when I forgot the '.0' part